IN THE CLAIMS

Please rewrite claims 1-14 and add new claims 15-21 as follows:

1. (Presently Amended) A software analysis tool comprising:

means for converting software entities and their relationships into a graph having a structure of nodes interconnected by edges, said graph further comprising a tree comprising a plurality of subtrees, each said subtree representing one or more nodes in the graph, the relationship among the plurality of subtrees representing the edges among nodes in the graph; and

an editor comprising means for allowing a user to edit the graph.

~~wherein the graph includes a meta node and edge representing a child graph, and~~

wherein the software entities comprise software program code.

2. (Original) A software analysis tool as claimed in claim 1, wherein the conversion means comprises means for bi-directionally folding and unfolding a graph between meta and child levels.

3. (Original) A software analysis tool as claimed in claim 1 or 2, wherein the editor comprises means for automatically generating fresh graph layouts after manipulation.

4. (Previously Presented) A software analysis tool as claimed in claim 1 or 2, wherein the conversion means comprises a plurality of back-ends, each being associated with an aspect of a software system comprising software program code.

5. (Original) A software analysis tool as claimed in claim 4, wherein each back-end comprises means for converting the entities and the relationships of the associated aspect into nodes and edges of the graph.

6. (Previously Presented) A software analysis tool as claimed in claim 4, wherein the back-ends are associated with managers.

7. (Original) A software analysis tool as claimed in claim 6, wherein the managers comprise means for routing commands between the editor and the back-ends.

8. (Previously Presented) A software analysis tool as claimed in claim 6, wherein each manager is associated with a group of back-ends associated with a group of back-ends.

9. (Original) A software analysis tool as claimed in claim 8, wherein the back-ends associated with a particular manager share a common interface and set of operations.

10. (Cancelled)

11. (Presently Amended) A dependency analysis system recorded on a computer-readable medium, comprising:

a node class for instantiating node objects in memory representing aspects of an analyzed system as nodes of a graph having a structure of nodes interconnected by edges, said graph further comprising a tree comprising a plurality of subtrees, each said subtree representing one or more nodes in the graph, the relationship among the plurality of subtrees representing the edges among nodes in the graph;

a connection class for instantiating connection objects in memory representing dependencies between aspects of the analyzed system;

an edge class for instantiating edge objects representing collections of one or more connections or edges, wherein said analyzed system comprises software program code.

12. (Original) The dependency analysis system of claim 11, further comprising:

at least one subclass of the node class, the subclass being specific to a particular category of system.

13. (Original) A dependency analysis system recorded on a computer-readable medium, comprising:

an abstraction layer for providing a uniform interface to third-party analysis tools;

a graph model data structure for storing dependency information derived through the abstraction layer from third-party tools; said graph model structure comprising a structure of nodes interconnected by edges, said graph model structure further comprising a tree comprising a plurality of subtrees, each said subtree representing one or more nodes in the graph, the relationship among the plurality of subtrees representing the edges among nodes in the graph,

a rendering system for providing a plurality of views of the graph model data structure.

14. (Previously Presented) A dependency analysis system comprising:

a data structure stored in computer memory representing a hierarchy of graphs, each said graph having a structure of nodes interconnected by edges, said graph further comprising a tree comprising a plurality of subtrees, each said subtree representing one or more nodes in the graph, the relationship among the plurality of subtrees representing the edges among nodes in the graph;

a rendering system for displaying the hierarchy of graphs;

a user interface responsive to a user action indicating a command to expand a displayed node, the user interface causing the rendering system to replace the displayed node with one or more embedded child nodes in response to the user action.

15. (New) The dependency analysis system of claim 14, wherein the data structure further comprises a one to many mapping from a first directed graph to a second directed graph, wherein every element in the first directed graph corresponds to exactly one element in the second directed graph, and any element in the second directed graph corresponds to one or more elements in the first directed graph.

16. (New) The dependency analysis system of claim 15, where the the data structure further comprises a one to many mapping from the second directed graph to a third directed graph, wherein every element in the second directed graph corresponds to exactly one element in the third directed graph, and any element in the third directed graph may correspond to one or more elements in the second directed graph.

17. (New) The software analysis tool of claim 1, wherein the graph further includes a meta node and edge representing a first child graph, said first child graph further comprising a meta node and edge representing a second child graph.

18. (New) The dependency analysis system of claim 13, wherein the graph model data structure further comprises a one to many mapping from a first directed graph to a second directed graph, wherein every element in the first directed graph corresponds

to exactly one element in the second directed graph, and any element in the second directed graph may correspond to one or more elements in the first directed graph.

19. (New) The dependency analysis system of claim 18, wherein the graph model data structure further comprises a one to many mapping from the second directed graph to a third directed graph, wherein every element in the second directed graph corresponds to exactly one element in the third directed graph, and any element in the third directed graph may correspond to one or more elements in the second directed graph.

20. (New) A system for representing the relationship among elements of a complex system, comprising:

a tree compring a plurality of subtrees, each said subtree comprising a root node and one or more nodes; and

a relationship among the root nodes of said subtrees, said relationship including a dependency relationship;

wherein for each pair of said subtrees not sharing any common node, there exists a relationship between the root nodes of said pair of subtrees, if there exists a relationship between a node in one subtree of said pair of subtrees and a node in the other subtree of said pair of subtrees.

21. (New) The system of claim 20, wherein the complex system is a software system comprising a plurality of software entites and the relationship includes reference dependency among the software entites.